RFS-G90G93750(X)+ Programming Manual

AN-60-146

www.minicircuits.com

Table of Contents

1 Introduction

This document provides detailed information about the use of application and system-level commands supported by the firmware of the Mini-Circuits RFS-G90G93750(X)+ high power signal source / power amplifier. The RFS-G90G93750(X)+ is a solid state connectorized 750W signal source and amplifier module which can be used in a wide range of industrial, scientific, and medical applications in the 902-928 MHz frequency range.

The ISC, RFS, and RFX series can be operated by sending text commands over their serial interface(s). These commands are part of a non-proprietary command protocol made to be both legible by humans and suitable for process automation through software. The serial command set enables users to get started quickly and communicate directly with RFS-G90G93750(X)+ using nothing more thana standard USB cable and PC. Though this document is specific to the RFS-G90G93750(X)+, most of the commands and functions described in this document are shared between products in the ISC, RFS, and RFX series.

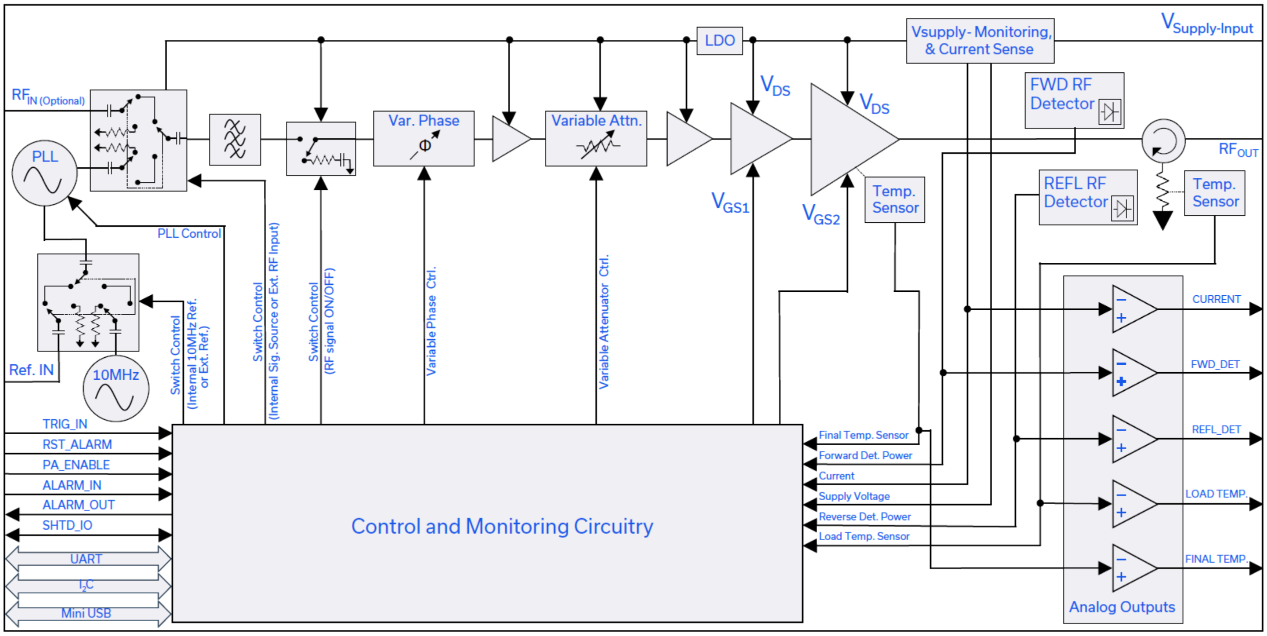
1.1. Style Conventions

This document uses the following style conventions:

Monospace	Denotes text that is entered at a keyboard, such as commands, file names, and source code.
[bracketed]	Denotes a named argument in a command.
(bracketed)	Denotes text that is optional or an optional argument in a command.

1.2. System Overview

BLOCK DIAGRAM



Functional Description

The RFS-G90G93750(X)+ is a full RF Energy (RFE) Generator Subsystem incorporating a PLL frequency synthesizer, Input Switch, Variable Attenuator, Power Amplifier, Circulator, and Control/Monitoring/Protection Circuitry in a single compact module. All functions needed to monitor and control the RFS-G90G93750(X)+ are accessible through the serial command set.

A serial connection can be made over either the 3.3V UART pins on the "CTRL1" 30-pin connector, or the Mini-USB "CTRL2" connector. The same text-based serial commands are used to communicate with the module for both USB and UART mode and only one communication mode can be active at a time with the default being USB. Refer to the $COMS command in this document for switching between USB and UART communication modes. In addition to the serial interface, the RFS-G90G93750(X)+ offers several functions through dedicated pins on the "CTRL1" 30-pin connector. A full description of these pins is outside of the scope of this document.

1.3. Setting Up Communication

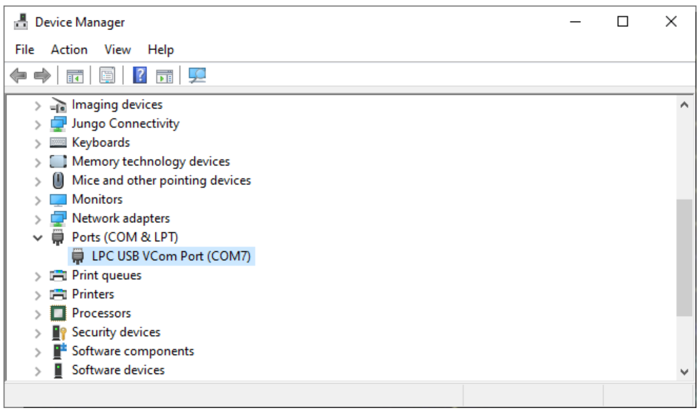
Setting up the communication between user and RFS-G90G93750(X)+ board over USB is a straightforward process. Below is an example using PuTTY on a Windows or Linux PC. Equivalent terminal emulator software may be used as well.

USB Serial Connection with PuTTY

- Plug the RFS-G90G93750(X)+ into the PC using a USB A to USB Mini-B cable.
- The RFS-G90G93750(X)+ will appear as a virtual COM port. Find the port name of the device.

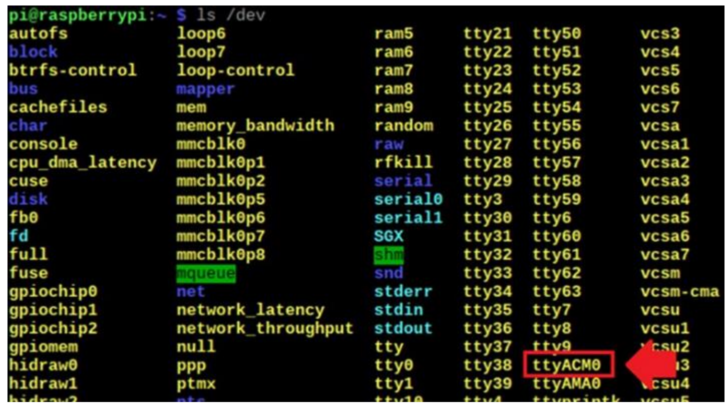
Windows:

Open the ‘Device Manager’ in Windows and find the port name of your device. It will show up as “LPC USB VCom Port”, followed by the port name.



Linux:

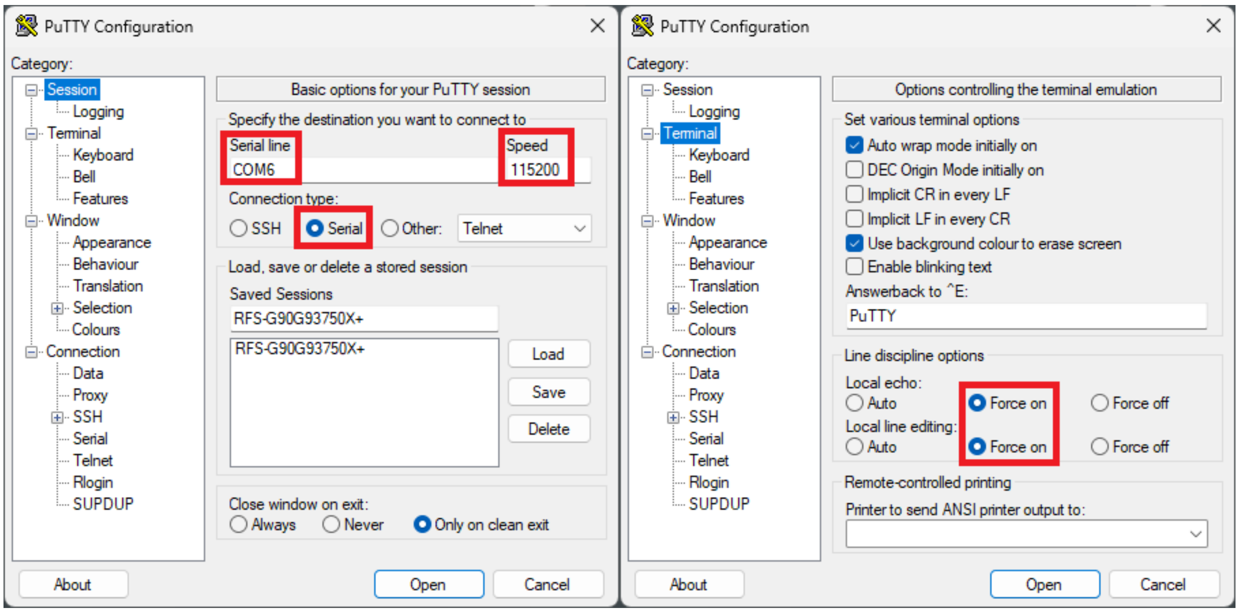
Open a terminal (CTRL + ALT + T) and use the "ls /dev" command to view available devices. The RFS-G90G93750(X)+ should appear as a "ttyACM" device followed by a number.



- Open PuTTY on your PC and provide the necessary information for the connection with the device.

Baud Rate:	115200
Data Bits:	8
Parity Bits:	0
Stop Bite:	1
Flow Control:	none

- It is highly recommended to configure the “Terminal” category settings: “Local echo” and “Local line editing” to “Force on”.
- Remark: Linux requires the full path to the port (e.g. “/dev/ttyACM0”).



- Save the session, so that it won’t need to be reconfigured again in the future and press ‘Open’ to start a connection with the RFS-G90G93750(X)+.
- A blank terminal window will pop up. Communication with the RFS board should now be established.

Sending a Command

The command set can be divided into three categories:

- **Set commands –** These are used to write the setpoints and configurations of the RFS board and typically end with the letter ‘S’. For example: setting the RF power level (PWRS), enabling DLL mode (DLES), etc.
- **Get commands –** These are used to read the values of the RFS board and, with a few exceptions, typically end with the letter ‘G’. For example: printing out setpoints (PWRG), getting the temperature (PTG), etc.
- **Execute commands –** A few remaining commands that don’t fall in either of the above categories. They will usually execute actions such resetting the board (RST) or performing a frequency sweep (SWP).

The general syntax for a command is as follows:

$[command],[channel],(parameter1),(parameter2),(...)\r\n

The protocol uses the ‘$’ symbol as an indicator for the start of a command. A message sent without the ‘$’ symbol will not be recognized as a command. Similarly, ‘\r\n’ (new line) marks the end of a command. Without at least a ‘\r’ or a ‘\n’ the device will keep waiting for more bytes indefinitely.

There is one parameter that is used in every command: the channel identifier.

Each RFS is assigned a numeric channel identifier. The default value is 1, but when using more than one RFS in a multi-channel setup, it may be desirable to assign a unique number to each device beforehand so that they can be identified and differentiated during runtime.

When sending a command, it is necessary to include the correct channel number of the board. Otherwise, the message is ignored under the assumption that it is intended for a different device.

The channel '0', is accepted by every RFS device regardless of the assigned number. In single-channel systems, as well as in multichannel systems where each RFS device has a dedicated serial connection, it is sufficient to always use channel identifier '0' when sending a command. In multi-channel systems that broadcast serial commands to multiple units over a bus, care should be taken to use the correct channel identifier.

1.4. Receiving a Response

The RFS board provides feedback when a command succeeds. A successful set command will always reply with an OK:

$[command],[channel],OK\r\n

A successful get command will simply return the requested information:

$[command],[channel],[parameter1],(parameter2),(...)\r\n

Miscellaneous commands do not have a standardized way of relaying successful results but will in most cases be similar to the above examples or some combination of both.

All command response strings will include the name of the command sent and the channel id of the device sending the response. If a command is sent to channel '0', the response will return with the channel the connected device is set to, never channel '0'.

It is also possible for the execution of a command to fail. There can be several reasons for this:

- A mistake could be made when writing out a command.
- The command may not be executable on the device type or configuration.
- A runtime problem occurs.

If something goes wrong during command execution, the RFS board will reply with an error code, indicating that the action has failed.

An error response looks as follows:

$[command],[channel],ERR##\r\n

The possible error codes and their respective descriptions are listed in the following table:

Hex Code	Error Description
0x01	Reserved
0x02	The serial message exceeded the maximum length.
0x03	The serial message had too few arguments.
0x04	The message had too many arguments.
0x05	The system could not accept this message is the current mode.
0x06	The system was busy and cannot process this message at this time.
0x07	The message was recognized but is not yet implemented in the codebase.
0x10	An argument was in error with the lower nibble indicating the argument number.
0x11	Argument 1 was invalid / out of range.
0x12	Argument 2 was invalid / out of range.

0x13	Argument 3 was invalid / out of range.
0x14	Argument 4 was invalid / out of range.
0x15	Argument 5 was invalid / out of range.
0x16	Argument 6 was invalid / out of range.
0x17	Argument 7 was invalid / out of range.
0x18	Argument 8 was invalid / out of range.
0x19	Argument 9 was invalid / out of range.
0x7E	Command execution failed.
0x7F	An error occurred that is not covered by any of the other error codes.

Table 1: Error Codes

An example of an error response would be the following:

Input:	$VER,1,1
Output:	$VER,1,ERR04

Hex code 0x04 indicates that the input command had too many arguments. This is correct, as the $VER command requires no additional arguments beyond the channel identifier.

1.5. Protocol

The protocol of the command set is straight forward:

1. Send a command.

2. Wait until a full response with a ‘\r\n’ at the end is read.

It is instrumental to always wait for a complete reply from the device before sending another command. Sending commands without waiting for a full reply can result in communication problems. Sending commands in quick succession without waiting for the response may additionally overburden the RFS board with a growing list of tasks to complete, leaving it no time to perform safety operations and resulting in an automatic reset to break the cycle.

2 Basic Functions

These are the essential commands core to microwave generator functionality.

2.1. $ECG - Get RF output enable state

This command returns the enable state of the RFS-G90G93750(X)+ RF output.

Syntax:

Input:	$ECG,[channel]
Output:	$ECG,[channel],[enable]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled, 1 = enabled

Example:

Input:	$ECG,1
Output:	$ECG,1,0

2.2. $ECS - Set RF output enabled / disabled

This command enables/disables the RF output on the RFS-G90G93750(X)+.

Syntax:

Input:	$ECS,[channel],[enable]
Output:	$ECS,[channel],[enable],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled, 1 = enabled

Example:

Input:	$ECS,1,1
Output:	$ECS,1,1,OK

2.3. $FCG - Get Frequency

This command reports the configured PLL synthesizer frequency in MHz.

Syntax:

Input:	$FCG,[channel]
Output:	$FCG,[channel],[frequency]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[frequency]** – Synthesizer frequency in MHz

Example:

Input:	$FCG,1
Output:	$FCG,1,915.5

2.4. $FCS – Set Frequency

This command configures the RF frequency of the PLL synthesizer in MHz. Valid frequency settings are in the range of 902-928MHz in steps of 0.5MHz.

Syntax:

Input:	$FCS,[channel],[frequency]
Output:	$FCS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[frequency]** – Synthesizer frequency in MHz.

Example:

Input:	$FCS,1,915.5
Output:	$FCS,1,OK

2.5. $PCG – Get Phase

This command reports the configured phase setting, in degrees, of the RFS-G90G93750(X)+.

Syntax:

Input:	$PCG,[channel]
Output:	$PCG,[channel],[phase]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[phase]** – Phase shifter setting in degrees

Example:

Input:	$PCG,1
Output:	$PCG,1,90.0

2.6. $PCS – Set Phase

This command configures the phase shifter setting, in degrees, of the RFS-G90G93750(X)+. Phase setting range is 0 to 360.

Syntax:

Input:	$PCS,[channel],[phase]
Output:	$PCS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[phase]** – Phase shifter setting in degrees

Example:

Input:	$PCS,1,45.0
Output:	$PCS,1,OK

2.7. $PIG – Read PA Current

This command returns total current, in amperes, from the drain power supply

Syntax:

Input:	$PIG,[channel]
Output:	$PIG,[channel],[current]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[current]** – total current in amperes

Example:

Input:	$PIG,1
Output:	$PIG,1,18.52

2.8. $PPDG – Get PA forward and reflected power in dBm

This command reports forward and reflected power in dBm.

Syntax:

Input:	$PPDG,[channel]
Output:	$PPDG,[channel],[forward power],[reflected power]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[forward power]** – Forward RF power output in dBm
- **[reflected power]** – Reflected RF power in dBm

Example:

Input:	$PPDG,1
Output:	$PPDG,1,50.00000,20.00000

This example indicates an output power level of +50 dBm, with +20 dBm being reflected back; this corresponds to an S11 of -30dB.

2.9. $PPG – Get PA forward and reflect power in watts

This command reports forward and reflected power in watts.

Syntax:

Input:	$PPG,[channel]
Output:	$PPG,[channel],[forward power],[reflected power]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[forward power]** – Forward RF power output in watts
- **[reflected power]** – Reflected RF power in watts

Example:

Input:	$PPG,1
Output:	$PPG,1,250.00000,25.00000

This example indicates an output power level of 250 watts, with 25 watts being reflected back; this corresponds to 10% reflected power or a VSWR of approximately 1.9:1.

2.10. $PTG – Get PA Temperature

This command returns the temperature, in degrees C, of the temperature sensor nearest to the Power Amplifier output device.

Syntax:

Input:	$PTG,[channel]
Output:	$PTG,[channel],[Temperature]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Temperature]** – PA temperature in degrees C

Example:

Input:	$PTG,1
Output:	$PTG,1,25.7

2.11. $PTTG – Get PA Termination Temperature

This command returns the temperature, in degrees C, of the temperature sensor nearest to the forward and reflected power detectors and reflected power termination resistors.

Syntax:

Input:	$PTTG,[channel]
Output:	$PTTG,[channel],[Temperature]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Temperature]** – Termination temperature in degrees C

Example:

Input:	$PTG,1
Output:	$PTTG,1,25.7

2.12. $PVG – Read PA Voltage

This command reports the measured Drain supply voltage in volts.

Syntax:

Input:	$PVG,[channel]
Output:	$PVG,[channel],[VDS]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[VDS]** – Measured Drain voltage in volts

Example:

Input:	$PVG,1
Output:	$PVG,1,50.10

2.13. $PWRDG – Get PA output power setpoint in dBm

This command reports the programmed desired output of the RFS-G90G93750(X)+ in dBm when in signal source mode.

Syntax:

Input:	$PWRDG,[channel]
Output:	$PWRDG,[channel],[power]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[power]** – programmed target output power in dBm

Example:

Input:	$PWRDG,1
Output:	$PWRDG,1,50.00

2.14. $PWRDS – Set PA output power setpoint in dBm

This command configures the programmed desired output of the RFS-G90G93750(X)+ in dBm when in signal source mode.

Syntax:

Input:	$PWRDS,[channel],[power]
Output:	$PWRDS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[power]** – Desired output power in dBm

Example:

Input:	$PWRDS,1,50.0
Output:	$PWRDS,1,OK

2.15. $PWRG – Get PA output power setpoint in watt

This command reports the programmed desired output of the RFS-G90G93750(X)+ in watts when in signal source mode.

Syntax:

Input:	$PWRG,[channel]
Output:	$PWRG,[channel],[power]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[power]** – Programmed desired output power in watts

Example:

Input:	$PWRG,1
Output:	$PWRG,1,500.0

2.16. $PWRS – Set PA output power setpoint in watts

This command configures the programmed desired output of the RFS-G90G93750(X)+ in watts when in signal source mode.

Syntax:

Input:	$PWRS,[channel],[power]
Output:	$PWRS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+

Example:

Input:	$PWRS,1,100.0
Output:	$PWRS,1,OK

3 Information Request

Request information about the RF Generator Board.

3.1. $IDN – Get identity

This command reports identity information (manufacturer, model, and serial number) for the amplifier.

Syntax:

Input:	$IDN,[channel]
Output:	$IDN,[channel],[manufacturer],[model],[serial number]

- **[channel]** – Channel Identification Number of
- **[manufacturer]** – Manufacturer of the RFS-G90G93750(X)+ (namely, Mini–Circuits)
- **[model]** – The model, namely, RFS-G90G93750(X)+
- **[serial number]** – The unique serial number of the RFS-G90G93750(X)+

Example:

Input:	$IDN,1
Output:	$IDN,1,Mini-Circuits,RFS-G90G93750(X)+,MD00003A2342

3.2. $RTG – Report Up Time

This command returns the up time of the Amplifier since its initialization. The uptime count restarts when the board is reset.

Syntax:

Input:	$RTG,[channel]
Output:	$RTG,[channel],[up time]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[up time]** – The up time in seconds since the last reset

Example:

Input:	$RTG,1
Output:	$RTG,1,601

3.3. $TCG – Get Microcontroller Temperature

This command returns the temperature, in degrees C, of the microcontroller's internal temperature.

Syntax:

Input:	$TCG,[channel]
Output:	$TCG,[channel],[Temperature]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Temperature]** – Microcontroller temperature in degrees C

Example:

Input:	$TCG,1
Output:	$TCG,1,25.7

3.4. $VER – Get version

This command returns the firmware version number.

Syntax:

Input:	$VER,[channel]
Output:	$VER,[channel],[manufacturer identifier],[major revision], [minor revision],[Build],[date stamp],[time stamp]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[manufacturer identifier]** – Firmware developer identifier.
- **[major revision]** – The version's major revision number.
- **[minor revision]** – The version's minor revision number.
- **[build]** – The version's build number.
- **[date stamp]** – The date on which the firmware was compiled.
- **[time stamp]** – The time at which the firmware was compiled.

Example:

Input:	$VER,1
Output:	$VER,1,Mini–Circuits,3,5,0,April 14, 2025,11:53:00

4 Pulse-Width Modulation

The RFS-G90G93750(X)+ has the capability to produce a PWM signal when configured as a signal source. It also can read the forward power and reflected power during the pulse when configured as a signal source or when configured as a stand-alone PA using an external PWM input signal.

When configured as a signal source, the module can generate a modulated signal with no external RF or trigger inputs. However, the internal RF signal source can also be modulated by an external trigger, TRIG_IN, on pin 30 of Crtl1. Use the $ETS command to enable this feature. When configured as a stand-alone amplifier with an external PWM RF input signal a trigger, TRIG_IN, is required on pin 30 of Ctrl1 to accurately read the forward and reflect power during the pulse. This feature is enabled using the $ETSS command.

Pulse Width Modulation allows the user to modulate the RF signal, and in turn the average power output of the system, by turning the signal ON and OFF at a set rate.

The following two parameters are used to achieve this:

- PWM frequency – Dictates how often the signal switches between ON and OFF.
- PWM duty cycle – Dictates the time ratio between ON and OFF each period.

Depending on the duty cycle, the average power output of the system will decrease to a percentage of its set output. For example, a 50% duty cycle at 750W results in an average RF power output of 375W. To ensure sufficient measurement time of the RF signal, the pulses generated by any PWM scheme must not be too short. Hence, the permissible PWM frequency and duty cycle are dependent on each other. The PWM frequency can vary between 1000 Hz – 19800Hz. To ensure accurate power readings (and therefore accurate power output), the minimum value of the duty cycle changes along with the PWM frequency according to the following formula:

DCmin = ROUNDUP (fPWM * TminPW / 10,000)

Where:

- DCmin is the minimum duty cycle as a percentage.
- fPWM is the PWM frequency between 1000 and 19800 Hz.
- TminPW is the minimum pulse length in microseconds. The minimum pulse width to ensure sufficient measurement time of the RF signal is 50µs.

This means that at 1000 Hz, the minimum duty cycle is 5% and at 19800 Hz it is 99%. Going over this frequency value would effectively disable PWM, as the minimum duty cycle becomes 100%. The user needs to keep these limitations in mind – the system will not check for the limits. For a reasonable control range, PWM frequencies between 1 and 2 kHz are recommended.

4.1. $DCFS - set PWM frequency for RF switch

This command configures the frequency of the PWM switch in Hz.

Syntax:

Input:	$DCFS,[channel],[frequency]
Output:	$DCFS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[frequency]** – Desired PWM frequency for the PWM switch in Hz

Example:

Input:	$DCFS,1,2000
Output:	$DCFS,1,OK

4.2. $DCS - Set duty cycle

This command configures the duty cycle, in percent (%) of the PWM switch.

Syntax:

Input:	$DCS,[channel][duty cycle]
Output:	$DCS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[duty cycle]** – Duty cycle, 0% to 100%

Example:

Input:	$DCS,1,50
Output:	$DCS,1,OK

4.3. $DCG - Get duty cycle

This command reports the duty cycle, in percent (%) of the PWM switch as well as the PWM frequency. Reserved parameters are included for compatibility with legacy companion GUI applications.

Syntax:

Input:	$DCG,[channel]
Output:	$DCG,[channel],[frequency],[reserved],[trigger mode], [reserved],[reserved],[reserved],[reserved],[reserved],[duty cycle]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[frequency]** – The current PWM frequency.
- **[reserved]** – Reserved. Parameter should be ignored.
- **[trigger mode]** – The current operational mode of the PWM triggering.
  - 1 – Free running
  - **2 – Reserved. Parameter should be ignored.**
  - **3 – Reserved. Parameter should be ignored.**
- **[Reserved]** – Reserved. Parameter should be ignored.
- **[Reserved]** – Reserved. Parameter should be ignored.
- **[Reserved]** – Reserved. Parameter should be ignored.
- **[Reserved]** – Reserved. Parameter should be ignored.
- **[Reserved]** – Reserved. Parameter should be ignored.
- **[duty cycle]** – The current duty cycle percentage value.

Example:

Input:	$DCG,1
Output:	$DCG,1,2000,0,1,255,255,255,255,0.000000,58

5 DLL and Sweep

In RF energy systems, frequency adjustment is used to optimize power transfer into the load. There are two built-in frequency adjustment routines that help the user find, then switch to the optimal frequency.

The sweep function performs a sweep over a user defined frequency range and reports back the forward and reflected powers. The user can request that data be returned on all the frequency points or just request the data on the frequency point with the best forward to reflected power ratio. This document will use the terms Return Loss, S_{11}, and reflection coefficient interchangeably to denote the forward to reflected power ratio. The forward to reflected power ratio can be called Return Loss or S_{11}. The sweep function searches the entire range for the optimal S_{11}, so it will find the global minimum within the granularity of the step size. If a sweep is executed in the middle of a sensitive process, the fluctuating power delivered to the load caused by the sweep itself may not be tolerable, in which case, DLL mode may be preferred. The output of the sweep functions $SWP or $SWPD can be used to generate S_{11} plots similar to a network analyzer.

Digital Locked Loop (DLL) mode is another way to optimize frequency that is intended to be used during a process application. If the measured S_{11} is less than the threshold (poor match), then DLL will continuously sweep the frequency from 'lower frequency' to 'upper frequency' in steps of 'frequency step' until S_{11} exceeds the threshold (good match). While S_{11} exceeds the threshold, the DLL will use a local search strategy to fine-tune the frequency. Every step of the DLL, the current frequency is updated to the frequency with the best S_{11} of the three points (current frequency +/- frequency step). This way, the DLL can track optimal frequencies that move with changes in cavity temperature, pressure, or contents over the duration of a process application.

When DLL mode enabled, the only thing that changes operation-wise are the constant automatic adjustments to frequency setting, so it is possible to mix and match with other modes such as feed-forward mode, or There are no dropouts in power or fluctuations in frequency when the frequency is changed in the RFS-G90G93750(X)+, so it is appropriate for sensitive plasma applications.

5.1. $DLCG - get Digital Locked Loop (DLL) configuration

This command reports the configured parameters of the Digital Locked Loop (DLL) mode.

Syntax:

Input:	$DLCG,[channel]
Output:	$DLCG,[channel],[lower_frequency],[upper_frequency],[start_frequency],[step_frequency],[threshold],[main delay]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[lower frequency]** – The lower boundary of the bandwidth for DLL in MHz.
- **[upper frequency]** – The upper boundary of the bandwidth for DLL in MHz.
- **[start frequency]** – The frequency at which the DLL starts its activities in MHz.
- **[step frequency]** – The step size of the DLL in MHz.
- **[threshold]** – The match/efficiency threshold in dB to be met before DLL latches onto a frequency.
- **[main delay]** – The delay between complete runs of the DLL in ms.

Example:

Input:	$DLCG,1
Output:	$DLCG,1,902.0,928.0,902.0,1.0,10.0,100

5.2. $DLCS - set Digital Locked Loop (DLL) configuration

This command configures the parameters of the Digital Locked Loop (DLL) mode.

Syntax:

Input:	$DLCS,[channel],[lower_frequency],[upper_frequency],[start_frequency],[step_frequency],[threshold],[main delay]
Output:	$DLCS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[lower frequency]** – The lower boundary of the bandwidth for DLL in MHz.
- **[upper frequency]** – The upper boundary of the bandwidth for DLL in MHz.
- **[start frequency]** – The frequency at which the DLL starts its activities in MHz.
- **[step frequency]** – The step size of the DLL in MHz.
- **[threshold]** – The match/efficiency threshold in dB to be met before DLL latches onto a frequency.
- **[main delay]** – The delay between complete runs of the DLL in ms.

Example:

Input:	$DLCS,1,902,928,902,0.5,20,100
Output:	$DLCS,1,OK

5.3. $DLEG - Get Digital Locked Loop (DLL) Enable Status

This command reports whether the Digital Locked Loop (DLL) mode is enabled or disabled.

Syntax:

Input:	$DLEG,[channel]
Output:	$DLEG,[channel],[enable]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled, 1 = enabled

Example:

Input:	$DLEG,1
Output:	$DLEG,1,1

5.4. $DLES - Enable / Disable Digital Locked Loop (DLL)

This command ENABLES or DISABLES the Digital Locked Loop (DLL) mode.

Syntax:

Input:	$DLES,[channel],[enable]
Output:	$DLES,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled, 1 = enabled

Example:

Input:	$DLES,1,1
Output:	$DLES,1,OK

5.5. $SWP – Perform S11 sweep with RF output power in watts

This command performs an S11 frequency sweep across a band provided by the user. The output power during the sweep is provided in watts. Note that the input argument defining the output power level is in dBm.

Remark: The completion time of the command will increase as the number of frequency steps increases. This can make it seem as if the amplifier has become un–responsive for some time.

Remark: This command offers two output modes, which have different output syntaxes.

Syntax:

Input:	$SWP,[channel],[start frequency],[stop frequency],[step frequency],[power dBm],[output mode]
Output (Mode 0):	$SWP,[channel],[start frequency],[forward power], [reflected power] $SWP,[channel],[start frequency + step frequency],[forward power], [reflected power] $SWP,[channel],[start frequency + 2*step frequency],[forward power], [reflected power] o o o $SWP,[channel],[stop frequency],[forward power], [reflected power] $SWP,[channel],OK
Output (Mode 1):	$SWP,[channel],[frequency @ best match],[forward power], [reflected power]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[start frequency]** – The starting frequency of the sweep in MHz
- **[stop frequency]** – the ending frequency of the sweep in MHz
- **[step frequency]** – the step size in MHz
- **[power dBm]** – the output power, in dBm, at which the sweep is performed
- **[output mode]** – determines the format of the response.
  - Mode 0 is a verbose mode; in verbose mode, the frequency, measured forward power in watts, and measured reflected power in watts are reported for every measurement step in the sweep, followed by a SWP OK response.

- Mode 1 is a terse mode. In terse mode, one measurement result, specifically, frequency, forward power, and reflected power for the best match (i.e., lowest S11) of the sweep is reported in the response.

Example 1: Mode 0 (verbose mode)

This example executes a sweep from 902 MHz to 928 MHz in 2 MHz steps at 50 dBm (100W), and reports the result of each step. Finally, a "$SWP,1,OK" is returned, signifying the completion of the sweep.

Input:	$SWP,1,902,928,2,50,0
Output:	$SWP,1,902.0,100.013,8.873 $SWP,1,904.0,100.013,7.290 $SWP,1,906.0,100.013,5.878 $SWP,1,908.0,100.013,4.516 $SWP,1,910.0,100.013,3.639 $SWP,1,912.0,100.013,2.954 $SWP,1,914.0,100.013,2.596 $SWP,1,916.0,100.013,2.348 $SWP,1,918.0,100.013,2.606 $SWP,1,920.0,100.013,3.063 $SWP,1,922.0,100.013,4.078 $SWP,1,924.0,100.013,5.857 $SWP,1,926.0,100.013,8.553 $SWP,1,928.0,100.013,12.653 $SWP,1,OK

Example 2: Mode 1 (terse mode)

This example executes a sweep from 902 MHz to 928 MHz in 2 MHz steps at 50 dBm (100W), similarly to the previous example, but only the result of the best match, which occurred at 916 MHz, is returned. In Mode 1, the operating frequency and the DLL start frequency are automatically updated to the resulting best match frequency.

Input:	$SWP,1,902,928,2,50,1
Output:	$SWP,1,916,100.013,2.348

5.6. $SWPD – Perform S11 sweep with RF output power in dBm

This command performs an S11 frequency sweep across a band provided by the user. The output power during the sweep is provided in dBm.

Remark: This command is similar to $SWP with the exception that the forward and reflected power levels are reported in dBm instead of watts.

Syntax:

Input:	$SWPD,[channel],[start frequency],[stop frequency],[step frequency],[power dBm],[output mode]
Output:	[See $SWP command, paragraph 5.5]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[start frequency]** – The starting frequency of the sweep in MHz
- **[stop frequency]** – the ending frequency of the sweep in MHz
- **[step frequency]** – the step size in MHz
- **[power dBm]** – the output power, in dBm, at which the sweep is performed
- **[output mode]** – determines the format of the response.
    - Mode 0 is a verbose mode; in verbose mode, the frequency, measured forward power in watts, and measured reflected power in watts are reported for every measurement step in the sweep, followed by a SWP OK response.
    - Mode 1 is a terse mode. In terse mode, one measurement result, specifically, frequency, forward power, and reflected power for the best match (i.e., lowest S11) of the sweep is reported in the response.

Examples: [See $SWP command, paragraph 5.5]

6 Manual Power Control

In the default operation of the RFS-G90G93750(X)+, a power is set by the user with the $PWRS/$PWRDS command and the autogain feedback control loop will adjust the output power of the generator to minimize the difference between the power measured by the forward power detectors and the set power. The internal Digital Step Attenuator (DSA) 'Gain' component provides near-continuous fine-tuning adjustment over about a +/- 0.25 dB range. The autogain algorithm is constantly adjusting the DSA to match the power set by the user. In doing so, autogain compensates for system drifts caused by fluctuations in temperature, DC Voltage, or other factors.

Although it is not recommended for the general use-case, it is possible to disable the autogain feedback control loop and stop automatic updates of the DSA and bias settings. These settings can then be controlled manually. This manual power control mode is called Feed-Forward mode, as the feedback has been removed. In feed-forward mode, $PWRS/$PWRDS commands are still used to set the forward power, but their function changes. When a power is set using $PWRS/$PWRDS in feed forward mode, the DSA is set to the attenuation that would result in the requested power. The mapping between the power setting and the DSA setting is defined by the feed-forward calibration performed at the factory. The feed-forward calibration is defined over frequency and over power at a specified default PA bias setting at 50V, 25°C, into a matched load. The DSA attenuation and the PA bias setting states can also be changed directly or requested which can be useful for special cases or for system debugging.

6.1. $AGEG – Get Enable/disable Status of auto gain

This command returns the status of auto gain enable, i.e. whether or not the auto gain feature is turned on,

Syntax:

Input:	$AGEG,[channel]
Output:	$AGEG,[channel],[enable]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled, 1 = enabled

Example:

Input:	$AGEG,1
Output:	$AGEG,1,1

6.2. $AGES – Enable/disable auto gain

This command enables or disables the Auto Gain function in Signal Source mode. The Auto Gain function adjusts the digital step attenuator to accomplish a pre-programmed output power level (See $PWRDS and $PWRS commands, sections 0 and **Error! Reference source not found.**).

Syntax:

Input:	$AGES,[channel],[enable]
Output:	$AGES,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled, 1 = enabled

Example:

Input:	$AGES,1,1
Output:	$AGES,1,OK

6.3. $GCG – Get VGA attenuation in dB

This command reports the attenuator value of the digital step attenuator which sets the gain of RFS-G90G93750(X)+ in Amplifier Mode. The higher the value, the lower the gain, consequently the lower the RF output.

Syntax:

Input:	$GCG,[channel]
Output:	$GCG,[channel],[attenuation]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[attenuation]** – Attenuation setting of the digital step attenuator in dB

Example:

Input:	$GCG,1
Output:	$GCG,1,9.00

6.4. $GCS – Set VGA attenuation in dB

This command configures the attenuator value of the digital step attenuator which sets the gain of RFS-G90G93750(X)+ in Amplifier Mode. The higher the value, the lower the gain, consequently the lower the RF output.

Syntax:

Input:	$GCS,[channel],[attenuation]
Output:	$GCS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[attenuation]** – Attenuation setting of the digital step attenuator in dB

Example:

Input:	$GCS,1,9.00
Output:	$GCS,1,OK

6.5. $MCG – Get Magnitude in percent

This command has no effect on the RF behavior of the RFS-G90G93750(X)+, but the RFS-G90G93750(X)+ does respond with whatever value was assigned by the most recent $MCS command for compatibility with legacy companion GUI programs that require a response to $MCG commands.

Syntax:

Input:	$MCG,[channel]
Output:	$MCG,[channel] ,[magnitude]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[magnitude]** – Value assigned by most recent issuance of $MCS

Example:

Input:	$MCG,1
Output:	$MCG,1,50

6.6. $MCS – Set Magnitude in percent

This command has no effect on the RF behavior of the RFS-G90G93750(X)+, but the RFS-G90G93750(X)+ does accept and store a value and respond with an "OK" response for compatibility with legacy companion GUI programs that require a response to $MCS commands.

Syntax:

Input:	$MCS,[channel],[magnitude]
Output:	$MCS,[channel], OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[magnitude]** – A whole number between 0 and 100

Example:

Input:	$MCS,1,50
Output:	$MCS,1,OK

7 External Triggering

When the RFS-G90G93750(X)+ is set to PWM mode, the ADCs are automatically triggered to sample the forward and reflected power during the ON time of the pulse. When in amplifier mode, it is possible to supply an externally modulated signal to RF IN. This requires that a TTL signal be sent to the RFS to trigger the forward and reflected power ADCs to sample when the RF is ON. It is also possible to modulate the RF signal (On/Off) via TTL input. In this case, the ADC sampling trigger and the modulation trigger are the same TTL signal to TRIG_IN. The external triggering commands in this section allow the configuration of the system in these three cases.

7.1. $ETG - External Trigger Get

The external trigger setting status. See $ETS for description of setting.

Syntax:

Input:	$ETG,[channel]
Output:	$ETG,[channel],[mode]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[mode]** – 0 = internal PWM, 1 = external TRIG IN

Example:

Input:	$ETG,1
Output:	$ETG,1,0

7.2. $ETS - External Trigger Set

The external trigger setting allows the TRIG_IN port to modulate the signal via “RF signal ON/OFF & PWM switch” on the RFS-G90G93750(X)+. When the setting is enabled, a TTL high on TRIG_IN will turn the switch on and a TTL low on TRIG_IN will turn the switch off. When the setting is disabled (default), the switch is controlled by the RF Enable or modulated internally as in PWM Mode. If $ETSS is also enabled, then the power measurements will be synchronized with the rising edge of TRIG_IN.

Syntax:

Input:	$ETS,[channel],[mode]
Output:	$ETS,[channel],OK

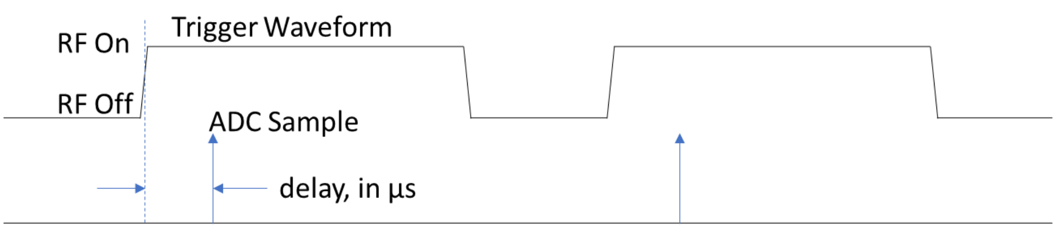
- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[mode]** – 0 = internal PWM, 1 = external TRIG IN

Example:

Input:	$ETS,1,0
Output:	$ETS,1,OK

7.3. $ETSDG - Get Trigger delay (us)

This command reports the delay time, in microseconds, from the rising edge of the RF switch control to the time the A/D converter takes a sample from the power detectors in synchronous ADC mode.



Syntax:

Input:	$ETSDG,[channel]
Output:	$ETSDG,[channel],[delay]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[delay]** – Delay in microseconds (refer to figure above)

Example:

Input:	$ETSDG,1
Output:	$ETSDG,1,100

7.4. $ETSDS - Set Trigger delay (us)

This command sets the delay time, in microseconds, from the rising edge of the RF switch control to the time the A/D converter takes a sample from the power detectors in synchronous ADC mode.

Syntax:

Input:	$ETSDS,[channel],[delay]
Output:	$ETSDS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[delay]** – Delay in microseconds (refer to figure in paragraph 0)

Example:

Input:	$ETSDS,1,100
Output:	$ETSDS,1,OK

7.5. $ETSG – Get External Trigger ADC Synchronization Enable Status

This command reports whether synchronous external ADC triggering is enabled or not. When synchronous ADC triggering is enabled, the sampling of the forward and reflected power detectors is done at a fixed delay (programmed by $ETSDS) after the leading edge of the EXT TRIG in waveform. Otherwise, the forward and reflected detectors are sampled asynchronously, as often as possible (approximately 1000 samples per second).

Syntax:

Input:	$ETSG,[channel]
Output:	$ETSG,[channel],[enable]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled (asynchronous), 1 = enabled (synchronous)

Example:

Input:	$ETSG,1
Output:	$ETSG,1,1

7.6. $ETSS – Set External Trigger ADC Synchronization Enable Status

This command configures sampling of the forward and reverse power detectors as synchronous (samples taken at a fixed delay after the rising edge of EXT TRIG) or asynchronous (as often as possible). Asynchronous sampling may prove useful in cases where the EXT TRIG waveform is a very wide pulse.

Syntax:

Input:	$ETSS,[channel],[enable]
Output:	$ETSS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[enable]** – 0 = disabled (asynchronous), 1 = enabled (synchronous)

Example:

Input:	$ETSS,1,1
Output:	$ETSS,1,OK

8 Safe Operating Area

The “Safe Operating Area” (SOA) feature defines limits on operating conditions for the RF generator/Amplifier which assure the reliability, longevity, and safe operation of the device. Operating the generator “outside” of the SOA limits could potentially damage the unit or at least reduce its expected lifetime. All Mini-Circuits generator devices will take self-protection actions that make them robust against accidental misuse.

There are 11 SOA types as shown in the below table. Most SOA types have “high” and “shutdown” limits defined. If a parameter exceeds the shutdown limit, RF will be disabled and the appropriate status register bit will be set to 1 (see $ST). The status bits will remain high until the next call to clear the errors (See $ERRC). If a parameter exceeds the warning limit, the appropriate status bit will be set to 1, but RF will remain enabled. For most SOAs, the purpose of the high limit is informational only. A warning can be displayed to notify the user that they are operating close to the limit.

SOA Type	Description	Enabled in RFS-G90G93750(X)+	Status Bit Warning Limit Exceeded	Status Bit Shutdown Limit Exceeded
0	Temperature SOA	Enabled	0x0000000002	0x0000000004
1	Internal Watchdog	Disabled	-	-
2	Reflection SOA	Enabled	0x0000000008	0x0000000010
3	External Watchdog	Disabled	-	-
4	Dissipation SOA	Disabled	0x0000080000	0x0000100000
5	PA status	Disabled	-	-
6	PLL lock lost	Disabled	0x0008000000	-
7	Current SOA	Enabled	0x0010000000	0x0020000000
8	Voltage SOA (Min)	Enabled	0x0200000000	0x0100000000
	Voltage SOA (Max)	Enabled	0x0400000000	0x0800000000
9	Forward Power SOA	Enabled	0x0040000000	0x0080000000
10	Termination Temperature SOA	Enabled	0x1000000000	0x2000000000

Table 2: SOA Status Bits

SOA settings in the RFS-G90G93750(X)+ are initialized in factory and can’t be modified by operator.

8.1. $SCG – report SOA drain current limit

This command reports the SOA drain current warning and fault limits.

Syntax:

Input:	$SCG,[channel]
Output:	$SCG,[channel],[Warning Limit],[Fault Limit]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Warning Limit]** – The maximum allowable drain current in Amperes before raising a Warning flag in the BIT status word.
- **[Fault Limit]** – The maximum allowable drain current in Amperes before shutting down the amplifier and raising a Fault flag in the BIT status word.

Example:

Input:	$SCG,1
Output:	$SCG,1,26,27

8.2. $SDG – Get dissipation SOA configuration

This command reports the SOA power dissipation warning and fault limits.

Syntax:

Input:	$SDG,[channel]
Output:	$SDG,[channel],[Warning Limit],[Fault Limit]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Warning Limit]** – The maximum power dissipation in watts before raising a Warning flag in the BIT status word.
- **[Fault Limit]** – The maximum power dissipation in watts before shutting down the amplifier and raising a Fault flag in the BIT status word.

Example:

Input:	$SDG,1
Output:	$SDG,1,1000,2000

8.3. $SFG – report SOA forward power limit

This command reports the SOA forward power warning and fault limits.

Syntax:

Input:	$SFG,[channel]
Output:	$SFG,[channel],[Warning Limit],[Fault Limit]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Warning Limit]** – The maximum allowable forward power in W before raising a Warning flag in the BIT status word.
- **[Fault Limit]** – The maximum allowable forward power in W before shutting down the amplifier and raising a Fault flag in the BIT status word.

Example:

Input:	$SFG,1
Output:	$SFG,1,775,800

8.4. $SOAGG – Get SOA grace timer

This command reports the grace timer value, in milliseconds, for the Safe Operating Area protection system. The grace timer value is the maximum amount of time, in milliseconds, during which a SOA limit violation can persist before the protection logic shuts the amplifier down.

Syntax:

Input:	$SOAGG,[channel]
Output:	$SOAGG,[channel],[grace]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[grace]** – maximum time, in milliseconds, for a SOA violation to persist before the amplifier is shut down

Example:

Input:	$SOAGG,1
Output:	$SOAGG,1,2000

8.5. $SOG – Get SOA configuration

This command reports the enable state of the SOA's protection systems.

8.5.1. $SOG Recommended Syntax:

The recommended $SOG syntax requests enable status of one individual SOA parameter.

Input:	$SOG,[channel],[soa type]
Output:	$SOG,[channel],[soa type],[enable]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[soa type]** – SOA Type. Used in the recommended syntax. The alternative syntax only supports SOA types 0 – 10.
  - 0 – Temperature SOA (See $STG)
  - 1 – Watchdog (this input is ignored, the MCU watchdog is always enabled)
  - 2 – Reflection SOA (See $SPG)
  - 3 – External Watchdog (not used in RFS-G90G93750(X)+)
  - 4 – Dissipation SOA (See $SDG)
  - 5 – PA status (not used in RFS-G90G93750(X)+)
  - 6 – IQ modulator IQ lock (not used in RFS-G90G93750(X)+)
  - 7 – Current SOA (See $SCG)
  - 8 – Voltage SOA (See $SVG)
  - 9 – Forward Power SOA (See $SFG)
  - 10 – Termination Temperature (See $STTG)
- **[enable] –** SOA Enable Status
  - 0 = OFF, 1 = ON

Example:

Input:	$SOG,1,1
Output:	$SOG,1,1,1

8.5.2. $SOG Alternative Syntax

The alternative $SOG syntax, leaving off the [SOA type] argument, requests enable statuses of SOA types 0 through 7 in a single response.

Note: Alternative syntax only supports SOA types 0 – 7. It does not support Voltage SOA, Forward Power SOA, or termination temperature SOA.

Input:	$SOG,[channel]
Output:	$SOG,[channel], [temperature enable],[reflection enable],[external watchdog enable],[dissipation enable],[pa status enable],[iq mod enable],[drain current enable]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+

  Note: Enable flags below are all 0 = OFF, 1 = ON

- **[temperature enable]** – Enable state of the temperature protection system.
- **[watchdog enable]** – not used / ignored
- **[reflection enable]** – Enable state of the RF power reflection protection system.

- **[external watchdog enable]** – not used
- **[dissipation enable]** – Enable protection against excess power dissipation ($P_{DISS} = P_{DC} – P_{RF}$)
- **[pa status enable]** – not used
- **[iq mod enable]** – not used
- **[drain current enable]** – Enable protection against excessive drain current

Example:

Input:	$SOG,1
Output:	$SOG,1,1,0,1,0,0,0,0,1

8.6. $SPG – Get reflected power SOA configuration

This command reports the SOA reflected power warning and fault limits.

Syntax:

Input:	$SPG,[channel]
Output:	$SPG,[channel],[Warning Limit],[Fault Limit]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Warning Limit]** – The maximum allowable reflected power in dBm before raising a Warning flag in the BIT status word.
- **[Fault Limit]** – The maximum allowable reflected power in dBm before shutting down the amplifier and raising a Fault flag in the BIT status word.

Example:

Input:	$SPG,1
Output:	$SPG,1,58,58.7

8.7. $STG – Get temperature SOA configuration

This command reports the SOA temperature warning and fault limits for the PA output device.

Syntax:

Input:	$STG,[channel]
Output:	$STG,[channel],[Warning Limit],[Fault Limit]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Warning Limit]** – The maximum allowable temperature in degrees Celsius before raising a Warning flag in the BIT status word.
- **[Fault Limit]** – The maximum allowable temperature in degrees Celsius before shutting down the amplifier and raising a Fault flag in the BIT status word.

Example:

Input:	$STG,1
Output:	$STG,1,75,90

8.8. $STTG – Get temperature SOA configuration

This command reports the SOA temperature warning and fault limits for the PA's internal termination.

Syntax:

Input:	$STTG,[channel]
Output:	$STTG,[channel],[Warning Limit],[Fault Limit]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Warning Limit]** – The maximum allowable temperature in degrees Celsius before raising a Warning flag in the BIT status word.
- **[Fault Limit]** – The maximum allowable temperature in degrees Celsius before shutting down the amplifier and raising a Fault flag in the BIT status word.

Example:

Input:	$STTG,1
Output:	$STTG,1,80,85

8.9. $SVG – Get Voltage Limits (V)

This command reports SOA fault and warning limits for the drain voltage power supply.

Syntax:

Input:	$SVG,[channel]
Output:	$SVG,[channel],[fault_LL],[warning_LL],[warning_UL],[fault_UL]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[fault_LL]** – lower fault limit – amplifier shuts down if drain voltage goes below this limit
- **[warning_LL]** – lower warning limit – amplifier does not shut down if drain voltage goes below this limit, but a warning is flagged in the amplifier's status word
- **[warning_UL]** – upper warning limit – amplifier does not shut down if this limit is exceeded, but a warning is flagged in the amplifier's status word.
- **[fault_UL]** – fault upper limit – amplifier shuts down if drain voltage goes above this limit

Example:

Input:	$SVG,1
Output:	$SVG,1,48.0,49.0,51.0,52.0

9 Error Handling

9.1. $ERRC - clear the BIT status word

An error on the RFS board is accompanied by an informative error code ($ST). To ensure the code can be viewed, it stays in memory until manually cleared away. For safety purposes, depending on the exact value, further RF output of the RFS board may be blocked for as long as that non-zero error status remains.

The $ERRC command clears the error state and resets the protective systems that impede the RFS board while an error is present.

The process of status checking and error clearing is the core of error handling. The error status of the RFS board is polled to retrieve current information about any errors that have occurred during operation. This information can be used to decide an appropriate response. After the problem has been resolved, the errors reported by the RFS board should be cleared to signal to the device that the error status is no longer actual.

A situation may occur where an error state appears to persist despite attempts to clear it. In fact, the error is getting cleared properly but is immediately reactivated again because the underlying cause of the error is still present.

Syntax:

Input:	$ERRC,[channel]
Output:	$ERRC,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+

Example:

Input:	$ERRC,1
Output:	$ERRC,1,OK

This clears the error state of the RFS board.

If the root cause of the problem has been resolved, the error code will be 0 the next time the error state of the RFS board is polled. If the root cause of the problem remains (e.g. the external shutdown remains triggered), the error state will keep reverting back to a non-zero value until it has been dealt with appropriately.

9.2. $ST – Get Status

The $ST command is used to monitor the status of the RFS board.

It returns hexadecimal codes that can be compared against bitmasks to provide an overview of the system status.

Error status:

RFS boards have a safety feature called the 'Safe Operating Area' (SOA). If a fault occurs during operation, the SOA raises an error and takes action to protect the system. This is indicated by a red LED on the board. An error on the RFS board is accompanied by an informative error code which can be used to trace the problem.

To ensure the error code can be viewed, it stays in memory until manually cleared away. Error codes that shutdown RF will set the enable state to 0 and will also block the RF power from being turned on again until the error is cleared with $ERRC. Clearing the error will not enable the generator, but it will allow the generator to be re-enabled as long as the error remains cleared. If the user wants to see the current status of the PA, they must clear the errors before requesting status. In some applications, it may be desirable to periodically clear errors so that error statuses displayed always reflect the current state of the system.

The following table provides an overview of the error status bitmask:

Bit #	Hex Value	Status Error Description	Shut Down Amplifier?
N/A	0x0000000000	No Error or Warnings	No
0	0x0000000001	Unspecified Error	Yes
1	0x0000000002	High PA Temperature	No
2	0x0000000004	Shutdown PA Temperature	Yes
3	0x0000000008	High Reflected Power	No
4	0x0000000010	Shutdown Reflected Power	Yes
5 to 18	0x000007FFE0	Reserved	N/A
19	0x0000080000	SOA High Dissipation	No
20	0x0000100000	SOA Shutdown Dissipation	Yes
21 to 25	0x0003E00000	Reserved	N/A
26	0x0004000000	Alarm In	Yes
27	0x0008000000	PLL Lock Lost	No
28	0x0010000000	SOA High Current	No
29	0x0020000000	SOA Shutdown Current	Yes
30	0x0040000000	SOA High Forward Power	No
31	0x0080000000	SOA Shutdown Forward Power	Yes
32	0x0100000000	SOA Shutdown Minimum Voltage	Yes
33	0x0200000000	SOA Low Voltage	No
34	0x0400000000	SOA High Voltage	No
35	0x0800000000	SOA Shutdown Maximum Voltage	Yes
36	0x1000000000	SOA Load Overtemp Warning	No
37	0x2000000000	SOA Load Overtemp Shutdown	Yes
39	0x4000000000	EEPROM CRC Warning	No
39	0x8000000000	EEPROM CRC Shutdown (Unrecoverable)	Yes

Table 3: Status Bits List

Note: Unless otherwise stated, all commands that turn RF output OFF are blocking. “Blocking” here means that the RF output is switched off and it remains off until the $ERRC command clears the error. Only then can the RF output be switched on again.

Syntax:

Input:	$ST,[channel]
Output:	$ST,[channel],[Status Word]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[Status Word]** – Hexadecimal BIT status word

Example:

Input:	$ST,1
Output:	$ST,1,0.0

10 System Configuration

System configuration commands define the system-level behavior of the RFS Board.

10.1. $CHANG - Get channel identification number

This command returns the channel number of a board.

Note: $CHANG syntax differs from other commands in that its input version does not include a channel number. $CHANG is intended for use determining the assigned channel of a PA whose channel is not known.

Syntax:

Input:	$CHANG
Output:	$CHANG,[channel]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+

Example:

Input:	$CHANG
Output:	$CHANG,1

10.2. $CHANS - Set channel identification number

This command assigns a new channel number of the RFS-G90G93750(X)+.

Syntax:

Input:	$CHANS,[present channel],[new channel]
Output:	$CHANS,[present channel],[new channel],OK

- **[present channel]** – Existing channel Identification Number of RFS-G90G93750(X)+
- **[new channel]** – The channel being re-assigned

Example:

Input:	$CHANS,1,2
Output:	$CHANS,1,2,OK

10.3. $COMG - Get Communication Interface

This command reports the active serial port, UART (3.3V TTL) or USB.

Syntax:

Input:	$COMG,[channel]
Output:	$COMS,[channel],[interface]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[interface]** – 1 = UART port, 2 = USB port

Example:

Input:	$COMG,1
Output:	$COMS,1,2

10.4. $COMS - Set Communication Interface

This command sets the communication interface to UART (3.3V TTL) or USB. Only one communication interface can be active at a time. The default communication interface is USB. If the user switches to UART by sending a $COMS,1,1 command, the USB serial port will no longer be active. There is no difference in serial command syntax between UART and USB communication interfaces. Holding the RXD line for three or more seconds will return the unit back to its default communication interface.

Syntax:

Input:	$COMS,[channel],[interface]
Output:	$COMS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[interface]** – 1 = UART port, 2 = USB port

Example:

Input:	$COMS,1,2
Output:	$COMS,1,OK

10.5. $CSG - Get clock source

This command reports the position of the PLL reference select switch – internal or external

Syntax:

Input:	$CSG,[channel]
Output:	$CSG,[channel],[clock source]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[clock source]** – 0 = Internal, 2 = External

Example:

Input:	$CSG,1
Output:	$CSG,1,0

10.6. $CSS - Set clock source

This command controls the reference select switch for the PLL frequency synthesizer. The reference select switch selects either an internal 10 MHz reference oscillator or an external reference.

Syntax:

Input:	$CSS,[channel],[clock source]
Output:	$CSS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[clock source]** – 0 = Internal, 2 = External

Example:

Input:	$CSS,1,0
Output:	$CSS,1,OK

10.7. $PODG – Get Power Offset in dB

This command reports a programmed power offset factor in dB. Power offset is used to de–embed the RF cable between the output connector of the amplifier and the input connector of the load. Power offset is described in more detail in paragraph 10.8.

Syntax:

Input:	$PODG,[channel]
Output:	$PODG,[channel],[offset]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[offset]** – Power offset in dB

Example:

Input:	$PODG,1
Output:	$PODG,1,3.25

10.8. $PODS – Set Power Offset in dB

This command programs a power offset factor in dB. The power offset factor is used to de–embed, i.e., correct for cable loss of, the RF cable between the output connector of the amplifier and the input connector of the load. Forward power measurements are decremented by the power offset value. Reflected power measurements are incremented by the same offset value. In effect, the RF measurement plane is moved from the output connector of the amplifier to the input connector of the load. Closed–loop power control, feed forward power control, and SOA limit checking are all based on the corrected power measurements.

Syntax:

Input:	$PODS,[channel],[offset]
Output:	$PODS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[offset]** – Power offset in dB

Example:

Input:	$PODS,1,3.25
Output:	$PODS,1,OK

10.9. $RFSG – Get RF Source Configuration

This command will return the current signal source configuration. The source can be configured to "Internal PLL Synthesizer" or "External RF Input".

Syntax:

Input:	$RFSG,[channel]
Output:	$RFSG,[channel],[source]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[source]** – 0 = Internal PLL Synthesizer (Normal), 1 = External RF Input (Amplifier Mode)

Example:

Input:	$RFSG,1
Output:	$RFSG,1,0

10.10. $RFSS – Set RF Source Configuration

This command configures the current signal source configuration. The source can be configured to "Internal PLL Synthesizer" or "External RF Input". When switching between signal sources, RF is automatically disabled, and must be re-enabled with the "ECS" command before resuming operation.

Syntax:

Input:	$RFSS,[channel],[source]
Output:	$RFSS,[channel],[source],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[source]** – 0 = Internal PLL Synthesizer (Normal), 1 = External RF Input (Amplifier Mode)

Example:

Input:	$RFSS,1,0
Output:	$RFSS,1,0,OK

10.11.$RST – Execute system reset

This command performs a full system reset. The serial communication session will terminate after this command is issued.

Syntax:

Input:	$RST,[channel]
Output:	$RST,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+

Example:

Input:	$RST,1
Output:	$RST,1,OK

10.12.$UARTG – Get UART baud rate

This command reports the baud rate of the UART port.

Syntax:

Input:	$UARTG,[channel]
Output:	$UARTG,[channel],[baud rate]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[baud rate]** – UART communication interface baud rate

Example:

Input:	$UARTG,1
Output:	$UARTG,1,115200

10.13.$UARTS – Set UART baud rate

This command sets the baud rate used for communication through UART. Any value can be entered, but unsurprisingly, ongoing communication will break the moment this value is changed.

Changing the baud rate affects communication speed. Lowering it can cause noticeable communication delay, while increasing it can speed up communication and leave a larger CPU time-slice for other tasks. However, setting the baud rate too high may cause communication issues to arise, as the UART transceivers have limitations.

Remark: This setting does not affect communication through USB, only through UART.

Syntax:

Input:	$UARTS,[channel],[baud rate]
Output:	$UARTS,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[baud rate]** – UART communication interface baud rate

Example:

Input:	$UARTS,1,115200
Output:	$UARTS,1,OK

11 EEPROM Commands[1]

In some applications, it may be useful to have the RFS-G90G93750(X)+ power up into a non-default state. The microcontroller in the RFS-G90G93750(X)+ has an integrated EEPROM that allows the user to define the power-up state of the device. To define a new power-up state in the EEPROM, first configure the device using the standard commands shown in Table 4, then commit them to EEPROM using the $SAV command. Commands are available to recall the saved state ($RCL) and reset the saved EEPROM state to factory defaults ($FRST).

Parameter	Command(s)
PA Enable State	ECS/ECG[2]
COM Interface (USB or TTL UART)	COMS/COMG
DLL Configuration	DLCS/DLCG
DLL Enable State	DLES/DLEG
RF Frequency	FCS/FCG
Magnitude Setting in Percent	MCS/MCG
Attenuator Value	GCS/GCG[3]
Power Set Point	PWRS/PWRDS/PWRG/PWRDG
Power Offset	PODS/PODG
Autogain Enable State	AGES/AGEG
Channel Number	CHANS/CHANG
Ext. Trigger Delay	ETSDS/ETSDG
Trigger Mode	ETS/ETG
Trigger Sync	ETSS/ETSG
UART Baud Rate	UARTS/UARTG
Phase	PCS/PCG
Mode (Amplifier vs. Signal Source)	RFSS/RFSG
PSU Enable State	PSUES/PSUEG
PSU Current Limit	PSUIS/PSUIG
PSU Voltage Setting	PSUVS/PSUVG

Table 4: Parameters stored in the EEPROM and set on Boot

The EEPROM has a finite number of write cycles before it becomes unreliable. This means, the functions in this section (especially $FRST and $SAV) should be used with care and never be introduced into an automated sequence where they could be called thousands of times daily. Doing so could lead to corrupted data and reduce the unit lifetime.

1 The functionality described in this section is applicable to firmware versions 4.0.0 and up. In previous versions, various parameters were saved to EEPROM every time they were modified. In versions 4.0.0 and above, active settings can be changed using the same commands, but they are not committed to EEPROM unless either a $SAV command or a $FRST command is sent. The purpose of the change is to reduce wear on the EEPROM, thereby improving the useful life of the RFS-G90G93750(X)+ signal source / amplifier.

2 If the device is configured to boot into “Signal Source Mode” (RFSS/RFSG), RF Output will always be disabled (ECS/ECG) at boot. For Amplifier mode, it is possible to configure the device to boot into either RF Output Enabled “On” or Disabled “Off” state.

3 The last GCS commanded setting is recalled on initialization.

11.1. $EFAIL_G – Get EEPROM Failure Flag

This command returns the state of the EEPROM failure flag, i.e., whether the unit has experienced an unrecoverable EEPROM failure. In the event of an unrecoverable failure, the unit must be returned to Mini-Circuits for repair or replacement.

Syntax:

Input:	$EFAIL_G,[channel]
Output:	$EFAIL_G,[channel], [status]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[status]** – 0 = OK, 1 = Unrecoverable Failure.

Example:

Input:	$EFAIL_G,1
Output:	$EFAIL_G,1,0

11.2. $FRST – Factory Reset

This command reverts the RFS-G90G93750(X)+ to its original factory settings. Any EEPROM changes made via the $SAV command are replaced with factory defaults.

Syntax:

Input:	$FRST,[channel]
Output:	$FRST,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+

Example:

Input:	$FRST,1
Output:	$FRST,1,OK

11.3. $RCL – Recall EEPROM configuration

This command reverts the RFS-G90G93750(X)+ to the most recently saved EEPROM configuration.

Syntax:

Input:	$RCL,[channel]
Output:	$RCL,[channel],OK

- **[channel]** – Channel Identification Number of the RFS-G90G93750(X)+

Example:

Input:	$RCL,1
Output:	$RCL,[channel],OK

11.4. $SAV – Save Configuration to EEPROM

This command saves the current configuration of the RFS-G90G93750(X)+ to nonvolatile EEPROM memory.

Syntax:

Input:	$SAV,[channel]
Output:	$SAV,[channel],OK

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+

Example:

Input:	$SAV,1
Output:	$SAV,1,OK

12 Low Level Functions

12.1. $DCAG - Get Snapshot of DCA setting

This command reports the raw DCA command most recently sent to the digitally controlled attenuator.

Syntax:

Input:	$DCAG,[channel]
Output:	$ DCAG,[channel],[attenuation code]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[attenuation code]** – most recent 7-bit attenuation code sent to the digitally controlled attenuator. Attenuator changes in 0.25dB steps per LSB.

Example:

Input:	$DCAG,1
Output:	$ DCAG,1,36

12.2. $PAG – Get Forward and Reflected Power in Raw ADC Counts

This command returns the forward and reflected powers of the PA in ADC counts. The exact ADCs from which the information is derived may differ per PA configuration.

Remark: The ADC measurements are averaged. The command returns the averaged value of 10 ADC measurements, which is why the output contains decimal values.

Syntax:

Input:	$PAG,[channel]
Output:	$PAG,[channel], [FWD power], [RFL power]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[FWD power]** – The RF power output of the PA in ADC counts.
- **[RFL power]** – The RF power reflected back into the PA in ADC counts.

Example:

Input:	$PAG,1
Output:	$PAG,1,507.50000,433.70000

12.3. $XADC – Get all ADC values

This command returns raw 12-bit ADC values. The range for these ADC values is 0-4095.

Syntax:

Input:	$XADC,[channel]
Output:	$XADC,[channel], [pa_temperature],[term_temperature],[fwd_pwr],[rev_pwr], [vds_sense],[idrain_sense],[spare1],[spare2]

- **[channel]** – Channel Identification Number of RFS-G90G93750(X)+
- **[pa_temperature]** – Raw ADC measurement of PA temperature
- **[term_temperature]** – Raw ADC measurement of reflected power termination temperature
- **[fwd_pwr]** – Raw ADC measurement of the forward power detector
- **[rev_pwr]** – Raw ADC measurement of the reflected power detector
- **[vds_sense]** – Raw ADC measurement of the drain voltage supply
- **[idrain_sense]** – Raw ADC measurement of the drain current supply
- **[spare1]** – Always Zero
- **[spare2]** – Always Zero

Example:

Input:	$XADC,1
Output:	$XADC,1,2786,1118,12,8,0,8,0,0

13 Revision History

Revision	Date	FW Version	Description
OR	08/01/2025	4.0.0	Release to Web. Added Description to EEPROM Section.
X1	04/15/2025	3.5.0	Preliminary Revision